CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test – I

Sub:	NETWORK SECURITY			Sec	A B C & D			Code:	10EC832
Date:	13/ 03 / 2018	Duration:	90 mins	Max Marks:	50	Sem:	VIII	Branch:	ECE

Complete solution with scheme of evaluation

		Marks	OBE CO	OBE RBT
1	Define passive and active security attacks. Discuss the functioning of the following attacks	02	CO1	L1
a.	Masquerade.	02	CO1	L1
b.	Replay.	02	CO1	L1
c.	Modification of messages.	02	CO1	L1
d.	Denial of service.	02	CO1	L1

Definition of all the security attack concepts. ***02X05=10***

SECURITY ATTACKS:

A useful means of classifying security attacks is in terms of passive attacks and active attacks. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.

The release of message contents is easily understood (Figure 1.1 a).A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

A second type of passive attack, traffic analysis, is subtler (Figure 1.1b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

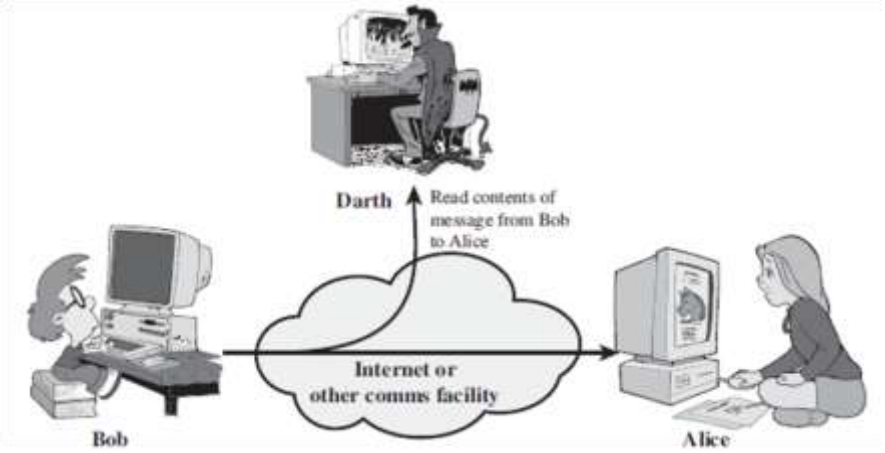
Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern.

However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

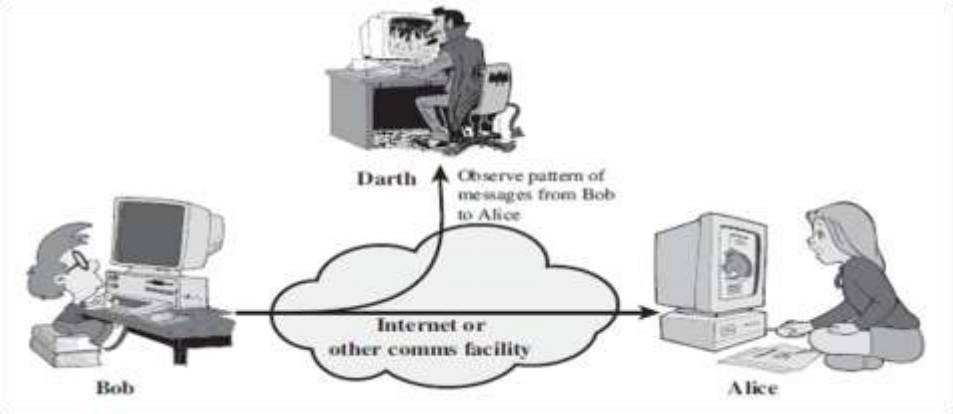
Active Attacks

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification Of messages, and denial of service.

A masquerade takes place when one entity pretends to be a different entity (Figure 1.3a). A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

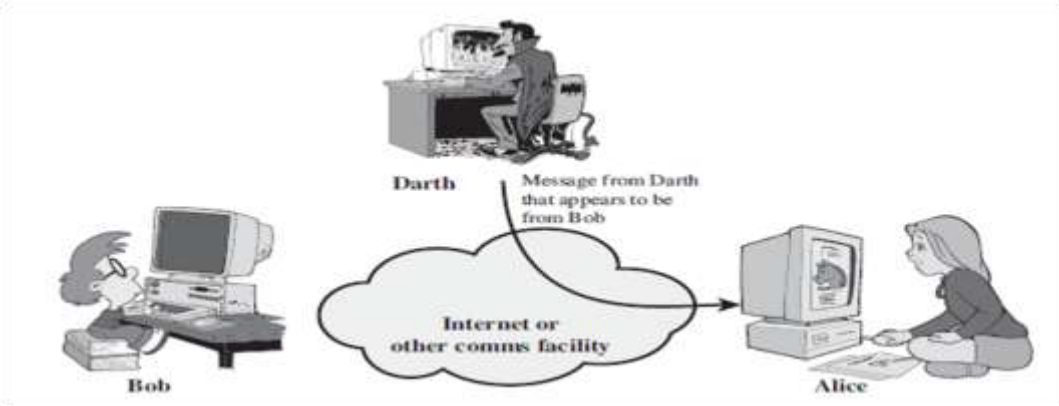


(a) Release of message contents



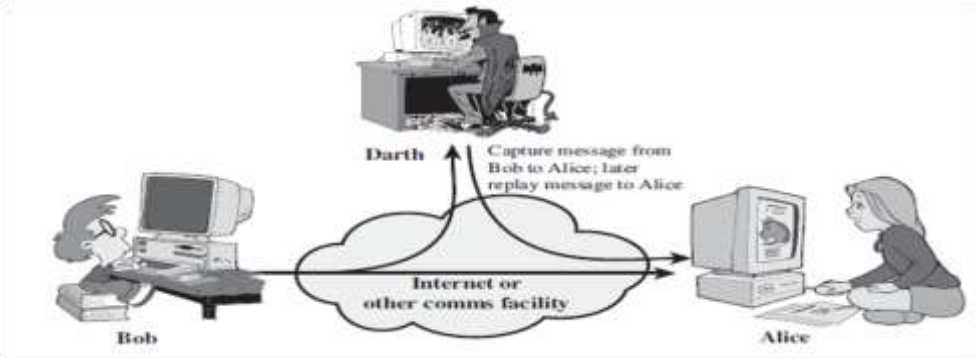
(b) Traffic analysis

Figure 1.1 Passive Attacks



(a) Masquerade

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (Figure 1.2 b).



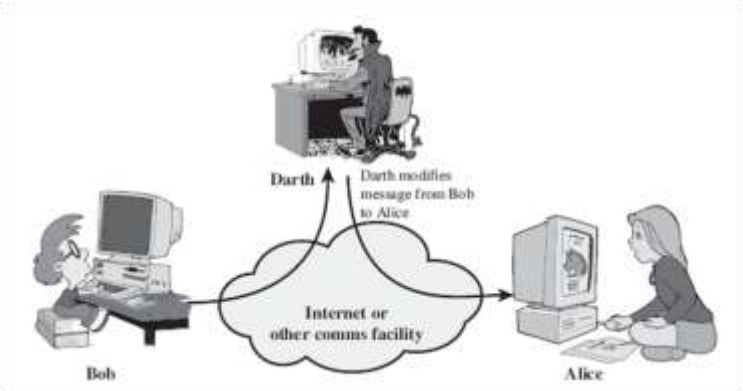
(b) Replay

Figure 1.2 Active Attacks

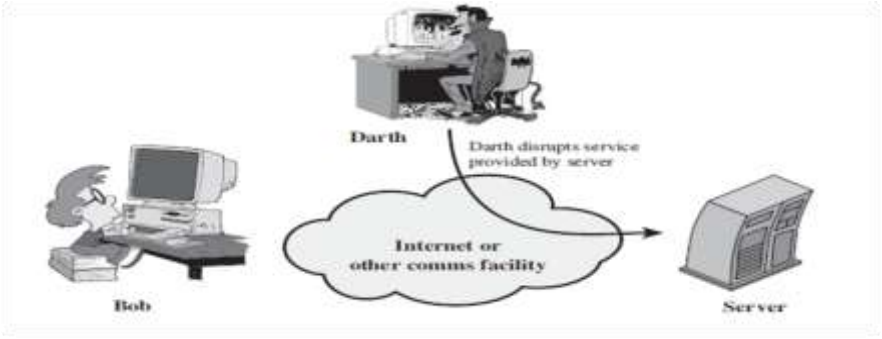
Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 1.3c). For example, a message meaning —Allow John Smith to read confidential file accounts‖ is modified to mean —Allow Fred Brown to read confidential file accounts.

The denial of service prevents or inhibits the normal use or management of communications facilities (Figure 1.3d). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination.

Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance. Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success.



(b) Modification of messages



(c) Denial of service

Figure 1.3 Passive Attacks

It is quite difficult to prevent active attacks absolutely because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.

2 Decrypt the cipher text "***BNCYZXQF***" using Hill cipher technique with key. $K = \begin{bmatrix} 3 & 4 \\ 19 & 11 \end{bmatrix}$. 10 | CO2 | L3

Calculation of $K^{-1} = \begin{bmatrix} 7 & 14 \\ 21 & 9 \end{bmatrix}$ ***05***

Obtaining pt : "hi my dear" ***05***

Given

Cipher Text: ***BNCYZXQF***

We need to decipher and obtain plain text

We know that

$$[pt] = [K]^{-1}[CT] \ mod\ 26$$

Given $K = \begin{bmatrix} 3 & 4 \\ 19 & 11 \end{bmatrix}$

We know that $[K]^{-1} = \frac{adj\ [K]}{|K|} \ mod\ 26$

Where

$$adj\ [K] = co\ factor[K]^T$$

$$co\ factor\ [K] = \begin{bmatrix} 11 & -19 \\ -4 & 3 \end{bmatrix} mod\ 26$$

$$\therefore adj\ [K] = \begin{bmatrix} 11 & -4 \\ -19 & 3 \end{bmatrix} mod\ 26$$

$$\therefore adj\,[K] = \begin{bmatrix} 11 & 22 \\ 7 & 3 \end{bmatrix}$$

$$|K| = \big((3 * 11) - (4 * 19)\big)\ mod\ 26$$
$$|K| = -43\ mod\ 26$$
$$|K| = -17\ mod\ 26$$
$$|K| = 9$$

Since $[K]^{-1} = \frac{adj\,[K]}{|K|}\ mod\ 26$

We have $[K]^{-1} = \frac{\begin{bmatrix} 11 & 22 \\ 7 & 3 \end{bmatrix}}{9}\ mod\ 26$

$$\rightarrow [K]^{-1} = \frac{1}{9}\begin{bmatrix} 11 & 22 \\ 7 & 3 \end{bmatrix} mod\ 26$$

Using extended Euclidian algorithm we can find the multiplicative inverse of 9mod26.

q	r_1	r_2	r	t_1	t_2	$t = t_1 - qt_2$
2	26	9	8	0	1	-2
1	9	8	1	1	-2	3
8	8	1	0	-2	3	-26
X	1	0	X	3	-26	X
since $r_2 = 0$, it indictes end of operation, and $t_1 = 3$						
$\therefore$ multiplicative inverse of 9 mod 26 is 3 mod 26						
$i.e., (9 * 3) = 1\ mod\ 26$						
$27\ mod\ 26 = 1\ mod\ 26$						

$$[K]^{-1} = \frac{1}{9}\begin{bmatrix} 11 & 22 \\ 7 & 3 \end{bmatrix} mod\ 26$$

Can now be re-written as

$$[K]^{-1} = 3 * \begin{bmatrix} 11 & 22 \\ 7 & 3 \end{bmatrix} mod\ 26$$

$$[K]^{-1} = \begin{bmatrix} 33 & 66 \\ 21 & 9 \end{bmatrix} mod\ 26$$

$$[K]^{-1} = \begin{bmatrix} 7 & 14 \\ 21 & 9 \end{bmatrix}$$

$$\therefore [pt] = [K]^{-1}[CT]\ mod\ 26$$

$$[pt] = \begin{bmatrix} 7 & 14 \\ 21 & 9 \end{bmatrix}\begin{bmatrix} B & C & Z & Q \\ N & Y & X & F \end{bmatrix}\ mod\ 26$$

$$= \begin{bmatrix} 7 & 14 \\ 21 & 9 \end{bmatrix}\begin{bmatrix} 1 & 2 & 25 & 16 \\ 13 & 24 & 23 & 5 \end{bmatrix}\ mod\ 26$$

$$= \begin{bmatrix} 189 & 350 & 497 & 182 \\ 138 & 258 & 732 & 381 \end{bmatrix}\ mod\ 26$$

$$= \begin{bmatrix} 7 & 12 & 3 & 0 \\ 8 & 24 & 4 & 17 \end{bmatrix}$$

$$= \begin{bmatrix} h & m & d & a \\ i & y & e & r \end{bmatrix}$$

*For the given cipher text : **BNCYZXQF***
*The plain text is: "**hi my dear**"*

3 Using play fair cipher, encipher the message "**bassoon guitar**" with key "**MONARCHY**". 10 CO2 L3

Obtaining the play fair table using the given key as

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

05

Obtaining the cipher text as : "IBXAPANAEWKSRM" 05

The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams The Playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers's Have His Carcase. In this case, the keyword is monarchy.

Obtaining the play fair table using the given key as

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. Plaintext is encrypted two letters at a time, according to the following rules:

1. *Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.*

2. *Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.*

3. *Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.*

4. *Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).*

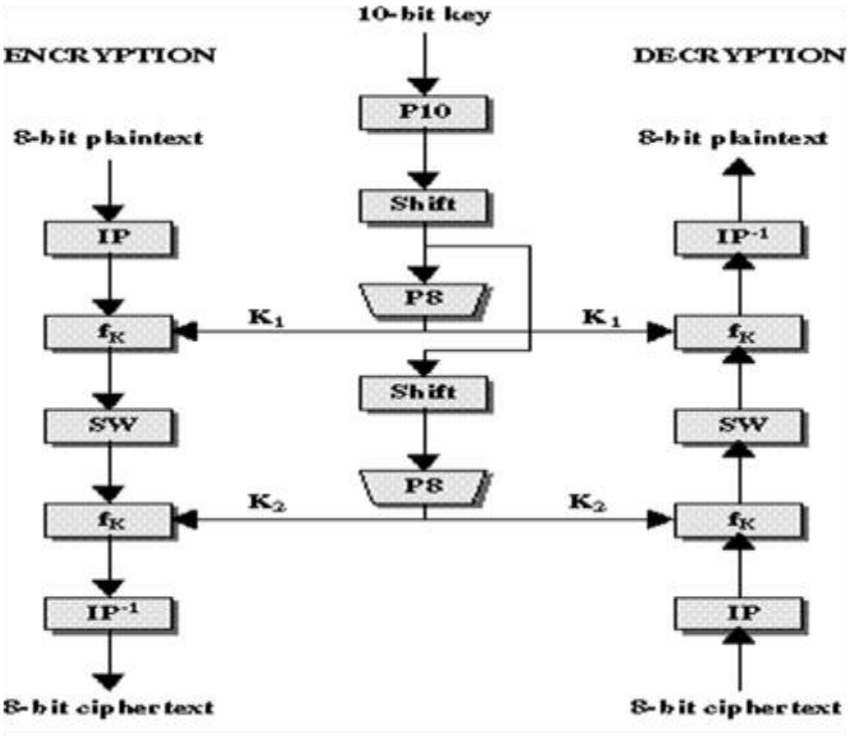
The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 x 26 = 676 digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time

considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

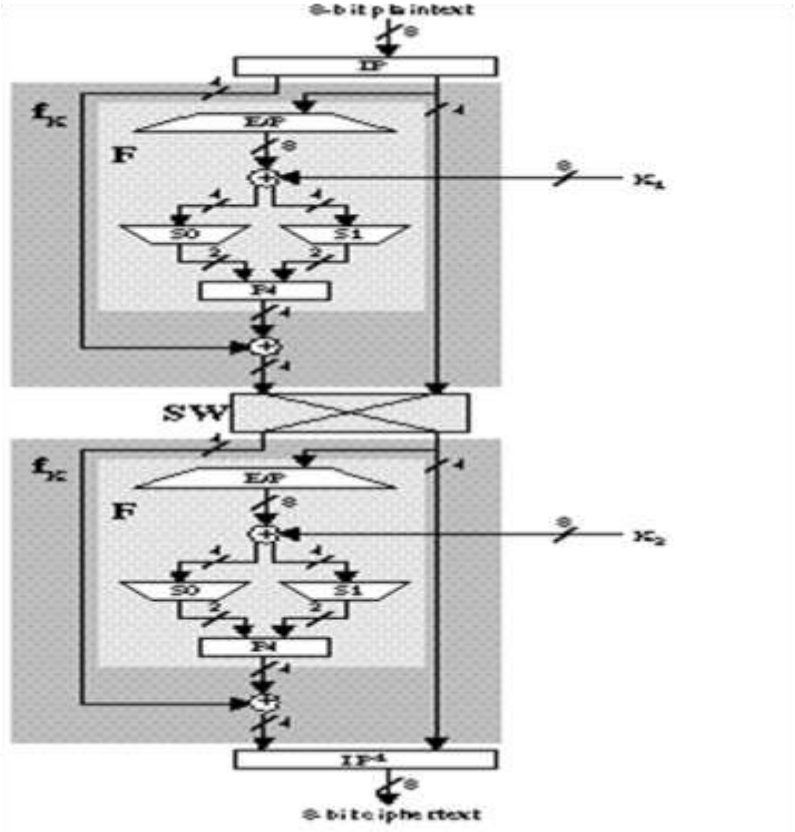
4	Explain SDES encryption/decryption and key generation using necessary diagrams.	10	CO2	L3

Neat diagram illustrating all the steps 04

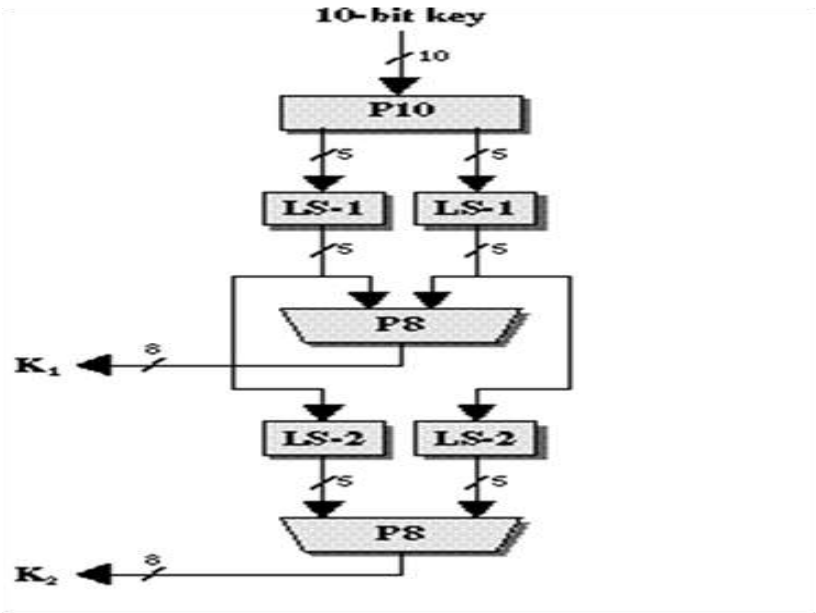
Explanation of encryption, decryption and key generation 06



Simplified DES algorithm



Encryption in detail



Key generation detail

Simplified DES, developed by Professor Edward Schaefer of Santa Clara University [SCHA96], is an educational rather than a secure encryption algorithm. It has similar properties and structure to DES with much smaller parameters.

Figure .1 illustrates the overall structure of the simplified DES, which we will refer to as SDES. The S-DES encryption algorithm takes an 8-bit block of

plaintext and a 10-bit key as input and produces an 8-bit block of ciphertext as output. The S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key used to produce that ciphertext as input and produces the original 8-bit block of plaintext. The encryption algorithm involves five functions: an initial permutation (IP); a complex function labeled fK, which involves both permutation and substitution operations and depends on a key input; a simple permutation function that switches (SW) the two halves of the data; the function fK again; and finally a permutation function that is the inverse of the initial permutation (IP-1). As was mentioned in Chapter 2, the use of multiple stages of permutation and substitution results in a more complex algorithm, which increases the difficulty of cryptanalysis. The function fK takes as input not only the data passing through the encryption algorithm, but also an 8-bit key. The algorithm could have been designed to work with a 16-bit key, consisting of two 8-bit subkeys, one used for each occurrence of fK. Alternatively, a single 8-bit key could have been used, with the same key used twice in the algorithm. A compromise is to use a 10-bit key from which two 8-bit subkeys are generated, as depicted in Figure G.1. In this case, the key is first subjected to a permutation (P10). Then a shift operation is performed. The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K1). The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K2).

We can concisely express the encryption algorithm as a composition1 of functions:

$$IP^{-1} \circ f_{K2} \circ SW \circ f_{K1} \circ IP$$

Which can also be written as:

$$ciphertext = IP^{-1}(f_{K2}(SW(f_{K1}(IP(plaintext)))))$$

where

$K1 = P8(Shift(P10(key)))$

$K2 = P8(Shift(Shift(P10(key))))$

Decryption is also shown in Figure .1 and is essentially the reverse of encryption:

$$plaintext = IP^{-1}(f_{K1}(SW(f_{K2}(IP(ciphertext)))))$$

We now examine the elements of S-DES in more detail

S-DES depends on the use of a 10-bit key shared between sender and receiver. From this key, two 8-bit subkeys are produced for use in particular stages of the encryption and decryption algorithm. Figure 2 depicts the stages followed to produce the subkeys.

First, permute the key in the following fashion.

Let the 10-bit key be designated as (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10). Then the permutation P10 is defined as:

P10(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6).

P10 can be concisely defined by the display:

P10									
3	5	2	7	4	10	1	9	8	6

This table is read from left to right; each position in the table gives the identity of the input bit that produces the output bit in that position. So the

first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on. For example, the key (1010000010) is permuted to (1000001100). Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits. In our example, the result is (00001 11000). Next we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:

P8							
6	3	7	4	8	5	10	9

The result is subkey 1 (K1). In our example, this yields (10100100) We then go back to the pair of 5-bit strings produced by the two LS-1 functions and perform a circular left shift of 2 bit positions on each string. In our example, the value (00001 11000) becomes (00100 00011). Finally, P8 is applied again to produce K2. In our example, the result is (01000011).

Figure 3 shows the S-DES encryption algorithm in greater detail. As was mentioned, encryption involves the sequential application of five functions. We examine each of these. Initial and Final Permutations G-5 The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function:

IP							
2	6	3	1	4	8	5	7

This retains all 8 bits of the plaintext but mixes them up. At the end of the algorithm, the inverse permutation is used:

IP^{-1}							
4	1	3	5	7	2	8	6

It is easy to show by example that the second permutation is indeed the reverse of the first; that is, IP-1(IP(X)) = X. The Function fK The most complex component of S-DES is the function fK, which consists of a combination of permutation and substitution functions. The functions can be expressed as follows. Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to fK, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings. Then we let f K(L, R) = (L ! F(R, SK), R) where SK is a subkey and ! is the bit-by-bit exclusive-OR function. For example, suppose the output of the IP stage in Figure G.3 is (10111101) and F(1101, SK) = (1110) for some key SK. Then fK(10111101) = (01011101) because (1011) ! (1110) = (0101). We now describe the mapping F. The input is a 4-bit number (n1n2n3n4). The first operation is an expansion/permutation operation:

E/P							
4	1	2	3	2	3	4	1

The first 4 bits (first row of the preceding matrix) are fed into the S-box S0 to produce a 2- bit output, and the remaining 4 bits (second row) are fed into S1 to produce another 2-bit output. These two boxes are defined as follows:

$$S_0 = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \qquad S_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

The S-boxes operate as follows. The first and fourth input bits are treated as a 2-bit number that specify a row of the S-box, and the second and third input bits specify a column of the Sbox. The entry in that row and column, in base 2, is the 2-bit output. For example, if (p0,0 p0,3) = (00) and (p0,1 p0,2) = (10), then the output is from row 0, column 2 of S0, which is 3, or (11) in binary. Similarly, (p1,0 p1,3) and (p1,1 p1,2) are used to index into a row and column of S1 to produce an additional 2 bits. Next, the 4 bits produced by S0 and S1 undergo a further permutation as follows:

P4			
2	4	3	1

The output of P4 is the output of the function F. The Switch Function The function fK only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits so that the second instance of fK operates on a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are the same. The key input is K2.

5 Explain the operation of Caesar Cipher with the algorithm. Decipher "***WKHTXIFNEURZQIRAMXPSVUYHUWKHODCBGRJ***" using Caesar Cipher. 10 CO2 L3

The English alphabet table

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Given cipher text table

W	K	H	T	X	I	F	N	E	U	R	Z	Q	I	R	A	M	X	P	S	V	U	Y	H	U	W	K	H	O	D	C	B	G	R	J
22	10	07	19	23	08	05	13	04	20	17	25	16	08	17	26	12	23	15	18	21	17	24	07	20	22	10	07	14	03	28	27	06	17	09

The plain text is given by:

$$pt = ct - key \bmod 26$$

For Caesar cipher the key=3

$$\therefore pt = ct - 3 \bmod 26$$

The plain text table for given cipher text

19	07	04	16	20	05	02	10	01	17	14	22	13	05	14	23	09	20	12	15	18	14	21	04	17	19	07	04	11	00	25	24	03	14	06
t	h	e	q	u	i	c	k	b	r	o	w	n	f	o	x	j	u	m	p	s	o	v	e	r	t	h	e	l	a	z	y	d	o	g

Obtaining the plain text as
"the quick brown fox jumps over the lazy dog" ***10***

6 Using SDES encipher the plain text "**00101000**" using the key "**1100011110**". Illustrating all the intermediate steps given as follows. 10 CO2 L3

P10	3	5	2	7	4	10	1	9	8	6
IP	2	6	3	1	4	8	5	7		
IP-1	4	1	3	5	7	2	8	6		
P8	6	3	7	4	8	5	10	9		
P4	2	4	3	1						

$$S_0 = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \qquad S_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

Given: pt = 00101000
Key = 1100011110

1. Key generation
K1

Bit #	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
P10(K)	0	0	1	1	0	0	1	1	1	1
shift(P10(k))	0	1	1	0	0	1	1	1	1	0
P8(shift(P10(k)))	1	1	1	0	1	0	0	1		

Key generation
K2

Bit #	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
P10(K)	0	0	1	1	0	0	1	1	1	1
Shift³(P10(k))	1	0	0	0	1	1	1	0	1	1
P8(shift³(P10(k)))	1	0	1	0	0	1	1	1		

So we have the two keys
K1 = 1110 1001
K2 = 1010 0111

2. Encryption
Given pt = 00101000

Bit #	1	2	3	4	5	6	7	8
pt	0	0	1	0	1	0	0	0
IP(Pt)	0	0	1	0	0	0	1	0
	Left(L)				*Right(R)*			
R	0	0	1	0				
E/P(R)	0	0	0	1	0	1	0	0
K1	1	1	1	0	1	0	0	1
E/P(R)⊕ K1	1	1	1	1	1	1	0	1
Sboxes(E/P(R)⊕ K1	1	0	0	0				
P4(Sboxes(E/P(R)⊕ K1)	0	0	0	1				
L	0	0	1	0				
P4(Sboxes(E/P(R)⊕ K1)⊕L	0	0	1	1				
Input to swap switch	0	0	1	1	0	0	1	0

Output from switch	0	0	1	0	0	0	1	1
	Left(L)				Right(R)			
R	0	0	1	1				
E/P(R)	1	0	0	1	0	1	1	0
K2	1	0	1	0	0	1	1	1
E/P(R)⊕ K2	0	0	1	1	0	0	0	1
Sboxes(E/P(R)⊕ K2	1	0	1	0				
P4(Sboxes(E/P(R)⊕ K2)	0	0	1	1				
L	0	0	1	0				
P4(Sboxes(E/P(R)⊕ K2)⊕L	0	0	0	1				
Input to IP^{-1}	0	0	0	1	0	0	1	1
Cipher text	1	0	0	0	1	0	1	0

7 a. Given cipher text "**NTSGYCNXEAKIETTHMAOT**" decrypt it using single round transposition technique if $K = 31452$. 5 CO1 L1

Obtaining the plain text as "enemy attacks tonight" **03**

Given

Cipher text = **NTSGYCNXEAKIETTHMAOT**

Encryption Key = **31452**

Decryption key generation

Standard	1	2	3	4	5
Encryption Key	3	1	4	5	2
Decryption Key	2	5	1	3	4

Decryption

2	5	1	3	4
N	Y	E	E	M
T	C	A	T	A
S	N	K	T	O
G	X	I	H	T

Rearrange to obtain

Plain text = **enemy attacks tonight**

b. Write a model for network security and explain it in details. 5 CO1 L1

Diagram of model for network security **03**

Explanation **02**

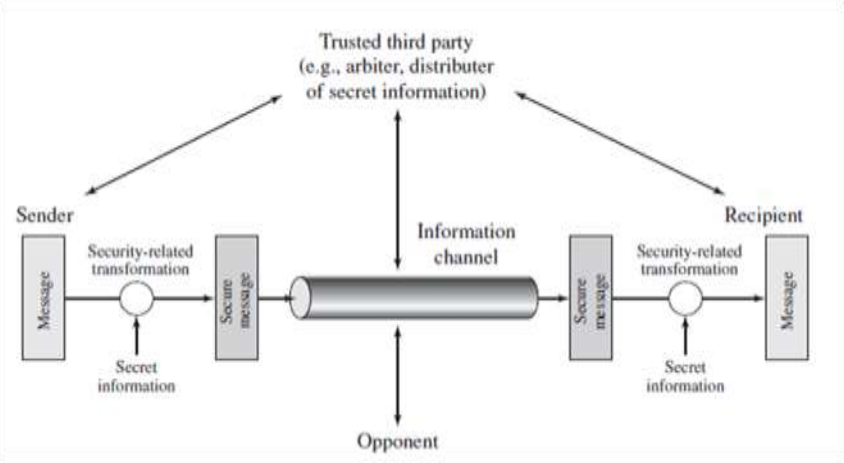


Fig : Model for network security

A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the principals in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.
Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of

the security algorithm and the secret information to achieve a particular security service.